# Serializer

El **componente Serializer** en **Symfony** es una herramienta poderosa para convertir objetos en formatos específicos (como XML, JSON o YAML) y viceversa. Sigue un esquema que utiliza un **array** como intermediario entre los objetos y su representación serializada.

1. **Normalizadores**: Los normalizadores se encargan de convertir objetos específicos en arrays y viceversa. El normalizador preferido es el **ObjectNormalizer**, pero hay otros disponibles.
2. **Encoders**: Los encoders transforman formatos específicos en arrays y viceversa. Por ejemplo, puedes usar **XmlEncoder** o **JsonEncoder**.

## Ejemplo de uso

use Symfony\\Component\\Serializer\\Encoder\\JsonEncoder;

use Symfony\\Component\\Serializer\\Encoder\\XmlEncoder;

use Symfony\\Component\\Serializer\\Normalizer\\ObjectNormalizer;

use Symfony\\Component\\Serializer\\Serializer;

$encoders = [new XmlEncoder(), new JsonEncoder()];

$normalizers = [new ObjectNormalizer()];

$serializer = new Serializer($normalizers, $encoders);

$person = new Person();

$person->setAge(30);

$person->setName('Alice');

// ... set other properties

$serializedData = $serializer->serialize($person, 'json');

//esto es para convertir ese objeto persona en un formato JSON

//este es un ejemplo al reves donde se obtiene unos datos en formato JSON y se convierte a un Objeto de tipo LowestPriceEnquiry

public function lowestPrice(int $id,Request $request,SerializerInterface $serializer): Response{  
  
 $deserializer=$serializer->deserialize($request->getContent(),LowestPriceEnquiry::class,'json');  
 dd($deserializer);  
 }

**3-Instalación**: Para usar el componente Serializer, instálalo con Composer: $ composer require symfony/serializer. Si lo instalas fuera de una aplicación Symfony, asegúrate de requerir el archivo vendor/autoload.php para habilitar la carga automática de clases.